Project 1

CSE 473/573 (Spring 2019)

Due Date: March 8th, 11:59 pm

1 Edge Detection [6 points]

The goal of this task is to experiment with two commonly used edge detection operator, i.e., Prewitt operator and Sobel operator, and familiarize you with tricks, e.g., padding, commonly used by computer vision practitioners. Specifically, the task is to detect edges in a given image, which is named "proj1-task1.jpg" and is stored in "`./data/`". You are required to implement all the functions that are labelled with "`# TODO`" in "task1.py" and "utils.py". In "task1.py" and "utils.py", we not only provide hints to you, but also provide utility functions that could be used as building blocks for you to complete this task. Therefore, you only need to write about 40 lines of code. Your code should be able to generate images that are identical to those stored in "`./results/`" (Note that images in "`./results/`" are provide to you for reference only. At test time, a different image will be used as input and the correct output images will be different from those already stored in "`./results/`").

Comment the lines "raise NotImplementedError" instead of deleting them, when you implement the functions labelled with "`# TODO`".

2 Character Detection [9 points]

The goal of this task is to experiment with template matching algorithms. Specifically, the task is to find a specific character in a given image, which is named "proj1-task2.jpg" and is stored in "`./data/`". You are required to implement a function named "detect" in "task2.py", which detects a character in an image. The function "detect" takes a given image and a given template that contains a character as inputs and returns the coordinates (i.e., coordinates of the top-left pixels) of the character contained in the template.

This task is composed of the following three sub tasks:

- Detect character "`a`" (lower case "`a`"). [3 point]
- Detect character "`b`" (lower case "`b`"). [3 point]
- Detect character "`c`" (lower case "`c`"). [3 point]

You need to customize your own templates. The templates containing character "`a`", "`b`" and "`c`" should be named as "`a.jpg`", "`b.jpg`", "`c.jpg`". They should be stored in "`./data/`".

Hints:

Using the image and template which only contain edges for template matching (detecting edges in both the image and the template) might give better results than using the original image and template for template matching, as edges only preserve the scales and shapes of characters, which are important for template matching. Distracting information, e.g., colors and fonts, of the characters are partially eliminated in the image and template that only contain edges. (Functions that we provide to you in "`utils.py`" and functions that you implement in "`task1.py`" could be used to detect edges.)

3 Guidelines

- Do **not** modify the code provided.
- Do **not** use any API provided by opencv (cv2) and numpy (np) in your code (except "np.sqrt()").
- Do **not** import any library (function, module, etc.).
- Ask for the TA's approval if you would like to use a programming language other than Python, C, C++, C# and Java.